

1. A method for the preparation and execution of a self-test procedure to validate the behavior of a processor model to be tested, the processor model being a processor or an associated simulator, wherein the method comprises the following steps, consisting in:

10 reading (E4), in a table, characteristic data
of the processor model to be tested, the data
comprising especially a functional definition of the
instruction to be tested and a functional definition of
the elements of the processor model,
15 computing an expected result (E6) at the end
of the execution of the instruction to be tested, the
computation being made from specifications of the user
and characteristic data of the processor model, and
 making the model to be tested carry out a
20 self-test procedure (E8) to validate the instruction to
be tested, the self-test procedure giving, in return, a
result word that is equal to a first value (OK) if the
behavior of the processor model is right, and equal to
a second value (ERROR) if the behavior of the processor
25 model is not right.

2. A method according to claim 1, wherein the self-test procedure (E8) comprises the following sub-steps, consisting in:

```

        initializing (E81) the elements of the
5  processor model to be tested,

```

executing the instruction to be tested (E82)
and obtaining a result,

comparing the obtained result (E83) and the
expected result, and

10 giving (E84) a word that is the result of the
comparison (OK/ERROR).

3. A method according to one of the claims 1
to 2, wherein at least two instructions of the set of
instructions of the processor model are executed
successively, to validate the performance
5 characteristics of the processor model to be tested.

4. A method according to claim 1, comprising
the following steps, consisting in:

making the model to be tested carry out a
self-test procedure (E8) and receiving a result word,

5 taking a decision (E10): if the above result
is equal to the first value, then performing a step
E12, else, if the above result is equal to the second
value (ERROR), carrying out a step E14,

taking a decision (E12): if another self-
10 test procedure has to be executed, then carrying out a
new step E8, if not, end of method, and

storing (E14) information on the self-test
procedure performed previously, the information
containing especially an address at which an error has
15 been detected, and then end of method.

5. A method according to claim 4,
furthermore comprising the following initialization
steps, to be executed before the first step E8:

09070424.053004
F00000" F220Z0860

6. A method according to one of the claims 4 or 5, furthermore comprising the following step E7, executed after the step E'6 and consisting in:

giving a statistical study (E7) of the

5 instructions to be tested during the following steps E8, to estimate the coverage of the set of instructions and performance characteristics of the processor model to be validated.

7. A method according to any of the claims 4 to 6, furthermore comprising the following step E16, performed at the end of the method and consisting in:

giving a statistical study (E16) of the

5 results (OK/ERROR) furnished during all the self-test procedures (E8) executed by the model being tested.

8. A method for the generation of self-test programs, comprising the steps E2 to E6 of the method according to one of the claims 1 to 7, and furthermore comprising the following step E'8:

5 writing (E'8) a self-test program to obtain the execution of a self-test procedure (E8) by the processor model.

9. A method according to claim 8 wherein,
during the step E'8, the self-test program is written
in an assembler type language, that can be understood
and executed by all the models to be tested of one and
5 the same processor.

10. A method according to one of the claims
8 or 9, wherein the step E81 is performed on the basis
of instructions of the set of the instructions of the
model to be tested.

11. A method according to one of the claims
8 to 10, implemented in the form of a program written
in an advanced DGL and/or C++ type language that can be
understood by the user.

00070424.053004